Lots o' Ticks: real-time high performance time series queries on billions of trades and quotes *

Arthur Whitney
KX Systems, Inc
www.kx.com
a@kx.com

Dennis Shasha
Courant Institute of Mathematical Sciences,
New York University,
http://cs.nyu.edu/cs/faculty/shasha/index.html
shasha@cs.nyu.edu

ABSTRACT

Financial mathematicians think they can predict the future by looking at time series of trades and quotes (called ticks) from the past. The main evidence for this hypothesis is that prices fluctuate only by a small amount in a given day and more or less obey the mathematics of a random walk. The hypothesis allows traders to price options and to speculate on stocks. This demonstration presents a query language and a parallel database (50-way parallelism) to support traders who want to analyze every tick, not just end-of-day ticks, using temporal statistical queries such as time-delayed correlations and tick trends. This is the first attempt that we know of to store and analyze hundreds of gigabytes of time series data and to query that data using a declarative time series extension to SQL (available at www.kx.com).

General Terms

time series, finance, parallel databases

1. QUERY LANGUAGE AND IMPLEMENTATION

The database management system KDB (produced by KX systems) is fully vertically partitioned and replaces tables by ordered arrays, which we call *arrables*. For this reason, moving average, correlation, and other such queries can be done directly in the database (see also [1]). Further, because the underlying programming language K shares the namespace of the database system, incorporating special purpose functions into the database language (called KSQL) entails only the overhead of a function call and each function can apply to one or more columns rather than a single row..

*Work partly supported by the US National Science Foundation grant 9988345.

ACM SIGMOD 2001 May 21-24, Santa Barbara, California, USA

The language KSQL is a semantic extension of SQL 92, though there are a few minor syntactic differences. The best way to understand KSQL is to look at a few examples.

1. Find the 5 tick moving averages (note the new keyword "avgs" with an s) of each stock per month, assuming that the trade arrable is ordered in ascending date order.

select 5 avgs price by stock, date.month from trade

(Stock and month are implicitly in the select clause. The result is a vector of moving averages associated with the ticks in each stock-month.)

2. Create in C (not shown) a function corr that takes two vectors as inputs and returns their correlation as output. Create in K an n-day delayed auto-correlation function. Then use it to find the 10 day delayed auto-correlation among stock prices, creating a new column ("auto10") on the table tradevec:

auto:{[n;x] corr[n drop x;-n drop x]}
update tradevec set auto10:auto[10,price]

2. DATA

The tick data concerns approximately 10,000 different securities – all of the NYSE, NASDAQ, American, and several other exchanges from 1993 to the present. This consists of several billions of trades and quotes, each about 40 bytes wide (i.e. a few hundred gigabytes). The hardware configuration consists of multiple Ultra SCSI II RAID or better devices and processors ranging from 500 MH to 1 GH. A single such machine can do time series and multi-dimensional aggregation on about 100 megabytes per second. Since KDB is column-oriented, this translates to 1-25 million rows per second depending on how many columns are involved in the analysis.

The data is naturally date and time ordered. The historical data is partitioned by date. Within each date, it is sorted by symbol (equity identifier) and time. The incoming data stream (up to 100,000 quotes and trades per second) is simply appended to a delta file in time order. At night, it is sorted by symbol and time and put into a partition which takes less than a minute. The incoming stream is logged.

3. REFERENCES

[1] Raghu Ramakrishnan, Donko Donjerkovic, Arvind Ranganathan, Keven S. Beyer, and Muralidhar Krishnaprasad: SRQL: sorted relational query language SSDBM 98